平台组件模块需求分析与概要设计

# 需求分析

首先对本文用到的一些概念给出定义，包括平台、组件、模型、算法。

平台：平台全称为“算法评估通用平台”。是指一个不依赖于具体项目，而提供通用的模型搭建，算法运行、验证等功能的平台。

组件：组件是某一个具体项目中涉及的具体事物。它包括两部分，一部分是集成在平台内而不依赖于具体项目，任何项目都会用到的内部组件，另一部分是某一具体项目要用到的具体事物。组件之间可以通过模型搭建进行组合，从而搭建出具体的模型。

模型：模型是一个组件的集合，以及组件之间的联系。它一般表示了一个具体应用的场景，而这一场景是我们进行算法运行、验证的具体目标。

算法：算法可以看做是一个特殊的组件。它通过对模型中的组件及其相互间的作用关系进行迭代，得到一组数据并对其进行分析，最终给出结果。

本文所说的平台组件模块，涵盖了全部上述四个内容。它是整个项目的核心部分。该模块的设计主要有以下几个方面的考虑。

## 通用性

通用性即为如何设计一个不依赖于具体项目的平台框架。之前的项目因为平台与外部组件部分并未做到抽象和剥离，导致在新的项目开发过程中，无法有效复用之前项目的通用部分，而需将前一个项目整体搬过来并逐处修改，或者直接重新推倒重建。很多时候是进行重复劳动。而贵实验室通过长期的项目积累和研究后发现，不同项目之间具有很大的相似性，完全可以通过一些合理的软件设计方法实现平台与外部组件部分的剥离。这样便可以在未来的项目中直接复用平台，而专注于特定组件的设计与开发。

本模块是该目标的直接体现。它需要将平台、内部组件和外部组件分别进行抽象、剥离和实现。使得平台不依赖于项目，内部组件与外部组件能够完美交互，外部组件独立于平台开发。在最理想的情况下，平台应该能够无需重新编译而兼容各种项目的外部组件。

## 针对特定项目的外部组件开发方法论

对于一个具体的项目，在复用平台及内部组件的前提下进行根据项目特定逻辑的外部组件的开发。因为此时平台已经和组件剥离，要让此项目的组件能够无差别的运行在该平台上，就必须遵循一定的方法论来进行组件的开发。这个方法论需要以文档、示例程序以等资源具体体现出来。本模块的实现主要参考贵实验室的飞行器测距项目进行研究。

在分析出该模块要点后，我们的需求可以分为以下几部分。

## 平台基础层

平台基础层主要包括组件行为的定义、内部组件的设计与实现，内部算法的加载与调用，外部组件的加载与交互。其中组件行为的定义主要是定义组件的一般行为，将其抽象成一组接口，平台对组件的操作均通过对接口的调用来实现，而组件的编程均通过对接口的实现来完成。内部组件主要包括要用到的向量类的设计与实现。内部算法的加载与调用通过对算法配置文件的读取，得到内部算法列表并且能够找到对应的Matlab编译出的DLL文件进行调用。外部组件的加载与交互通过对一个或多个预先编译好的外部组件DLL进行加载，并使之能与内部组件和算法进行交互。而这些要点的一个前提就是完全避免平台对于某一具体项目的重编译。

## 外部组件开发包

外部组件开发包主要包括将基础接口暴露，给出相关规范，供外部组件编写人员按照此规范进行外部组件的编写，还包括提供整个项目通用的一些工具类、函数和数据结构。

## 内部算法子模块

内部算法子模块主要包括通过对算法配置文件的读取获得算法的描述，包括算法ID，算法名称，算法DLL等信息，将这些信息注册到平台以便平台展示给用户供用户建模时选择。另外还要负责实现某一算法DLL的具体加载与调用，以供平台直接调用。

## 飞行器测距示例组件

以飞行器测距项目为例，设计与实现飞行器测距项目中的外部组件，供平台加载。

## 模型搭建程序

提供一个模型搭建程序，通过平台结合飞行器测距示例组件，能够对飞行器测距进行建模。

# 概要设计

该模块的设计主要是为了实现平台与组件之间的分离。为了实现这个目标，我们将平台和组件设计成通过接口来进行交互。接口即为平台定义的一组组件行为规范，即组件必须提供平台要求的这些功能，而平台所看到的组件，即为实现了这些接口的一些无差别的对象。对于外部组件之间，可以定义其特有的接口协议（即外部接口），外部组件之间通过外部接口进行交互，这一点对于平台来说不可见，平台可以不必关心外部接口间的具体业务往来，而只保证基础接口所提供的功能。这种关系可以用下图来描述。

平台

外部组件

外部组件

外部组件

外部组件

## 平台基础层

平台基础层的设计是为了提供一个统一的接口、组件管理模块，以供平台外程序，主要是服务器、客户端以及模型搭建程序等进行对平台功能的调用。这一部分将抽象出关于平台的全部对外功能，它主要包括以下几个方面：

### 从平台获得接口

该部分主要是平台提供对外的获得接口的功能。主要用到这一功能的是模型搭建程序。平台需要有办法将外部编写的组件库读入平台，并获得其定义的接口，并能够返回给外部程序。

### 从平台获得组件类型

该部分主要是平台提供对外的获得组件类型的功能。主要用到这一功能的是模型搭建程序。平台需要有办法将外部编写的组件库读入平台，并获得其定义的外部组件类型，并能够返回给外部程序。

### 从平台获得组件实例

该部分主要是平台提供对外的获得组件实例的功能。主要用到这一功能的是模型搭建程序。平台需要管理用户在模型搭建时创建的所有组件实例及其关联关系，并能够返回给外部程序。

## 平台与平台外组件之间的交互协议

平台基础层的设计主要是为了实现平台与组件之间的分离。为了实现这个目标，我们将平台和组件设计成通过接口来进行交互。接口即为平台定义的一组组件行为规范，即组件必须提供平台要求的这些功能，而平台所看到的组件，即为实现了这些接口的一些无差别的对象。

基础接口描述了组件的基本操作。平台将只可见组件的基础接口，并通过基础接口来操作组件。对于某个特定项目的所有组件，必须实现这组基础接口以供平台调用。其思路通俗的说，就是平台不关心某个项目的组件完成哪些具体功能（这些组件将自己实现这些功能），只知道这些组件遵循基础接口，那么便可通过基础接口无差别的操作这些组件。这组基础接口包括：

### 获得组件类型

通过该接口，平台可以知道这个组件类型。例：平台可通过这个接口知道飞行器组件的类型为“飞行器”，这样就可以在界面上显示出告之用户这个组件为飞行器。

### 获得组件属性

属性是指某个组件所包含的某个元素。例：飞行器会包含一个“运动”的属性来描述其运动方式。平台可以通过这个接口知道某个组件都有哪些属性，并呈现在界面上，告之用户这个组件有这样一个属性。

### 设置组件属性

平台可以通过这个接口实现对某个组件的某个属性的设置。例：可以根据用户的操作，将飞行器的运动属性设置为匀速直线运动或匀加速直线运动。又可以对匀速直线运动这个组件设置其速度或对匀加速直线运动设置其初速度及加速度。

### 检测组件合法性

合法性是指该组件目前被设置的状态是否合法。例：平台在用户建模完成并开始运行之前，需要检查飞行器的运动属性是否被设置。

以上即为几个基础接口，其余不再赘述，将会在最终的设计文档中做具体描述。平台为了能在运行时（而非编码和编译时）获知组件的信息，则需要在加载平台外组件时期完成对平台外组件的注册，而组件亦需要在运行时通过平台API获取平台的资源。这部分包括：

### 平台外组件注册

平台在加载平台外组件时，需要完成对平台外组件的注册，这里会通过调用平台外组件提供的回调函数完成。平台外组件在这时，在平台内注册自己的构造函数，属性列表及其它相关信息。

### 平台API

平台对外提供的一组函数。平台外组件如果需要平台提供的资源，比如对话框、内存、网络访问，可以通过这组API向平台申请。

## 平台外组件之间的交互协议

对于一个搭建好的模型，其中存在着多个外部组件实例，并且包含他们之间的关联关系。那么对于某一个外部组件，亦需要通过平台的功能，实现对于与其关联的组件的接口进行调用，实现他们之间的交互。这一部分的主要工作包括：

### 平台外组件的接口式编程

平台外组件为实现相互之间的交互，必须使用接口式编程方法。首先在设计平台外组件之前，必须定义出良好的接口，采用接口来抽象要操作的具体事物。接下来在组件设计时，需要对欲进行交互的另一组间进行类型检测（调用基础接口来实现），符合类型要求后，将目标组件转化成某一特定类型的接口，再调用该接口的具体某一方法。